Network Working Group

Request for Comment 147

NIC 6750

The Definition of a Socket

Joel M. Winett

Lincoln Laboratory

360/67

7 May 1971

Category: C1, C3, D1, H
RFC obsoleted: None
RFC updated: None
Related RFCs: RFC-129 (NIC-5845)

This material has not been reviewed for public release and is intended only for use with the ARPA network. It should not be quoted or cited in any publication not related to the ARPA network.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

TO: Network Socket Committee and Network Community 7 May 1971

FROM: J. M. Winett (LL)

SUBJECT: The Definition of a Socket

A socket is defined to be the unique identification to or from which information is transmitted in the network. The socket is specified as a 32 bit number with even sockets identifying receiving sockets and odd sockets identifying sending sockets. A socket is also identified by the host in which the sending or receiving processer is located.

Previous network papers postulated that a process running under control of the host's operating system would have access to a number of ports. A port might be a physical input or output device, or a logical I/O device supported by system calls to the host's operating system. The latter category includes a) I/O directed to a physical device which is being spooled by the operating system, b) a physical device whose basic characteristics have not been altered but whose access has been limited and possibly transformed by a mapping algorithm (e.g. device address mapping or cylinder relocation as in virtual mini disks), c) access to a file system through a directory and access method maintained by the operating system, d) a procedure for process to process communications, e) a procedure for machine to machine communication (such as defined by the network protocol.)

A socket has been defined to be the identification of a port for machine to machine communication through the ARPA network. Sockets allocated to each host must be uniquely associated with a known process or be undefined. The name of some sockets must be universally known and associated with a known process operating with a specified protocol. (e.g., a logger socket, RJE socket, a file transfer socket). The name of other sockets might not be universally known, but given in a transmission over a universally known socket, (e.g. the socket pair specified by the transmission over the logger socket under the Initial Connection Protocol (ICP). In any case, communication over the network is from one socket to another socket, each socket being identified with a process running at a known host.

The question arises as to whether socket names must be known to users of network programs or whether the specification of sockets can be made transparent to the user. Also, should the socket used at one time by a process be the same socket used at a later time by the same process for the same purpose? If sockets are not transparent to the user, then the sockets used must not be dependent on the order in which network connections are made.

The definition of a socket is also related to the accounting procedures followed for network usage. Network Control Programs (NCPs) should log each connection made and record the time the connection was made, the time the connection was closed, the number of messages and number of bits transmitted over the connection, the sending and receiving hosts, and the sockets at the sending host and receiving host which participated in the connection. In order for these sockets to be meaningful, they should be identifiable with the user, account, or process name with which each socket is associated.

It has previously been suggested that the sockets used by a network user be identified with that user no matter which host he used for network communications. Users would be registered at some host and be identified as a user from that host even if he used the system as a second host to communicate with the system at a third host.

To satisfy the above requirements within the name space of a 32 bit socket, the following procedure is suggested. (This procedure has been implemented with the NCP on the Lincoln Laboratory 360/67 system and is used by all processes making use of network facilities.) A 32 bit socket is divided into an 8 bit "home" field, a 16 bit "user" field and an 8 bit "tag" field. The tag consists of a 7 bit "plug" and a one bit "polarity" where a "0" polarity indicates a receive socket and a "1" polarity indicates a send socket. Thus a user on one host system may run processes with up to 128 send sockets and 128 receive sockets. This procedure allows for 256 hosts and 65,536 users per host. The only difficulty is in mapping user or process identifiers at a host into a 16 bit user number. This may be done through a table lookup, possibly using a hash coding technique. Though many systems have a unique run time index associated with each process, if this index were used as the user number, the user number would not be the same each time the process were used for network activity. What is required, is a unique mapping from a process identifier (usually a character string) into a 16 bit binary number.

The sockets used for facilities following a common network protocol, such as the ICP, should also follow this socket definition. Thus the logger socket at the Lincoln Laboratory 360/67 would be, and is, x'0A0000 01, ', i.e. home 10, user 0, and tag 1.

This procedure for defining sockets enables an accounting procedure for identifying users of network facilities and for measuring network usage.